

Enterprise Architecture – Designing the Assembly Line

In manufacturing we design a product and then implement an assembly line to deliver that product – why should it not be the same for knowledge workers?

This was the question we posed at the end of the first article (Enterprise Architecture - Moving to Structure that Delivers) in this two part series on enterprise architecture. In this second article, we take the proven principles of designing assembly lines from manufacturing and apply them to the knowledge worker assembly line.

To design the assembly line we must first establish our design principles.

What is it that we make? Many times I have asked this question of both business and IT to receive a series of blank and bemused looks. Whether it is the processing of additional contributions to superannuation, to managing a conference, to auditing accounts, the starting point is to know what knowledge workers make. This then begs the obvious of asking “by how many?”, “to what cost?”, “in what timeframe?” and “to what quality?”. If enterprise architecture is to ever deliver an assembly line that works, this most basic of information must be known and understood, i.e. enterprise architecture starts in the business with the business understanding their operations.

Every business activity is defined by a series of recipes. Follow the recipes and guarantee service delivery. This most basic of principle has been used by such diverse and iconic businesses as MacDonald’s and Ford to gain market dominance. We just need to develop the standardised recipes for knowledge workers and apply them, i.e. enterprise architecture must know the recipes to design the assembly line. Obvious.

An assembly line that reflects how the business operates has a greater chance of acceptance. This is yet another simple principle but is one that is missing in many enterprise architecture activities. Fact: we have an infinite ability to bypass that which does not work for us. For a knowledge worker assembly line to be designed and used, it must

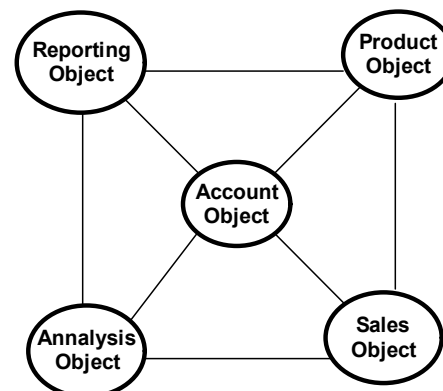
reflect the way the business operates. Without this, it will not be adopted. Yes, it may be wrong and yes we could do it better but remember, we can optimise it later and it is now the incumbent.

Now that we know what we make and how to make it, the next step is to design the assembly line to produce it to a consistent quality at controlled cost in a given time. This is done through user driven activity objects.

User driven Activity Objects define the assembly line.

The significance of defining the recipes to designing the knowledge worker assembly line is a given in manufacturing assembly lines. The additional advantage for that of the knowledge worker is the cascade from the system design through the required information management and work-flow and into the user interface provided.

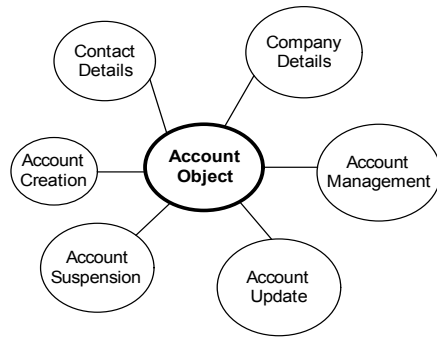
From each recipe (business activity), e.g. selling a product and the reporting on customer value, a series of activity objects can be defined.



Business activity defined by a series of business objects reflecting the way users work and the required standardised recipes.

Consider the account object. This would be composed of a series of other objects, e.g. company details and account creation details, i.e. define the recipe.

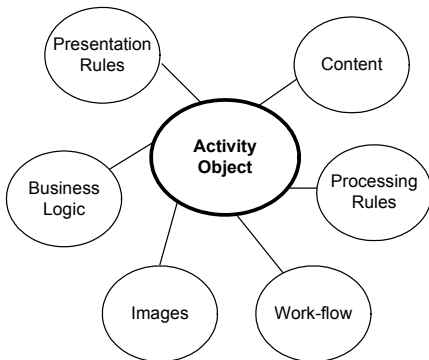
By using the recipes to define these objects and using a hierarchy of objects which reflect the way people work; the requirements, design, testing and optimisation of the knowledge worker assembly line becomes transparent - even to the most business illiterate systems developers.



For each object, a series of objects exist that encompass the required recipes.

Associated with each activity object are relationships and flows of information. Combine this with the required images, security, business logic and processing instructions; and a series of standard objects can be defined.

Since information, rules and processing logic etc. are abstracted from the source code (database driven), real time dynamic configuration, including rendering to an object driven interface (like touch screen menus on manufacturing assembly lines) ensures ready adoption by the knowledge worker.



Each activity object provides a framework for its database driven dynamical configuration with the required information, work-flow etc.

The net result of this approach is that business operations revolve around functional lines rather than the more traditional product lines. The advantages of this are self evident and include avoiding the duplication of operations and effort across business areas. In addition, a unified view of operations from either the business or customer perspective is achieved – with all that implies.

Enterprise architecture is really as simple as A B C:

- determine what it is you make,

- develop the required standardised recipes required to make it,
- use the recipes to drive out the standardised activity objects,
- use the activity objects to define the knowledge worker assembly line.

In our next article, we will review how to use out-sourcing and off-shoring in the context of the knowledge worker assembly line to further optimise and standardise business operations.

This document is for information purposes only and any advice given is of a general nature only and may not be applicable to an individual or a specific situation. IMS Corp. accepts no responsibility for any consequential loss or damage arising from the use of this document.

IMS Corp. is a specialist provider of services in the diagnosis, prescription and remedy of business problems. With a focus on empowering the client, IMS Corp. has met the needs of exceeded the expectation of clients to return shareholder value across the enterprise; to government, corporations and medium sized business alike.

IMS Corp.

Suite 5 275 Maroubra Road Maroubra NSW 2035 Australia

Email: info@imscorp.com.au

Web: www.imscorp.com.au

© Copyright IMS Corp.. All rights reserved.

All referenced trademarks are those of their respective owners.